

Protein Folding in the 2D Hydrophobic-Hydrophilic (HP) Square Lattice Model is Chaotic*

Jacques M. Bahi, Nathalie Côté, Christophe Guyeux, and Michel Salomon

August 23, 2016

Abstract

Among the unsolved problems in computational biology, protein folding is one of the most interesting challenges. To study this folding, tools like neural networks and genetic algorithms have received a lot of attention, mainly due to the NP-completeness of the folding process. The background idea that has given rise to the use of these algorithms is obviously that the folding process is predictable. However, this important assumption is disputable as chaotic properties of such a process have been recently highlighted. In this paper, which is an extension of a former work accepted to the 2011 International Joint Conference on Neural Networks (IJCNN11), the topological behavior of a well-known dynamical system used for protein folding prediction is evaluated. It is mathematically established that the folding dynamics in the 2D hydrophobic-hydrophilic (HP) square lattice model, simply called “the 2D model” in this document, is indeed a chaotic dynamical system as defined by Devaney. Furthermore, the chaotic behavior of this model is qualitatively and quantitatively deepened, by studying other mathematical properties of disorder, namely: the indecomposability, instability, strong transitivity, and constants of expansivity and sensitivity. Some consequences for both biological paradigms and structure prediction using this model are then discussed. In particular, it is shown that some neural networks seems to be unable to predict the evolution of this model with accuracy, due to its complex behavior.

1 Introduction

Proteins, polymers formed by different kinds of amino acids, fold to form a specific tridimensional shape. This geometric pattern defines the majority of functionality within an organism, *i.e.*, the macroscopic properties, function, and behavior of a given protein. For instance, the hemoglobin is able to carry oxygen to the blood stream due to its 3D geometric pattern. However, contrary to the

*Authors in alphabetical order

mapping from DNA to the amino acids sequence, the complex folding of this last sequence still remains not well-understood. Moreover, the determination of 3D protein structure from the amino acid linear sequence, that is to say, the exact computational search for the optimal conformation of a molecule, is completely unfeasible. It is due to the astronomically large number of possible 3D protein structures for a corresponding primary sequence of amino acids [1]: the computation capability required even for handling a moderately-sized folding transition exceeds drastically the computational capacity around the world. Additionally, the forces involved in the stability of the protein conformation are currently not modeled with enough accuracy [1], and one can even wonder if one day a fully accurate model can be found.

Then it is impossible to compute exactly the 3D structures of the proteins. Indeed, the Protein Structure Prediction (PSP) problem is NP-complete [2]. This is why the 3D conformations of proteins are *predicted*: the most stable energy-free states are looked for by using computational intelligence tools like genetic algorithms [3], ant colonies [4], particle swarm [5], memetic algorithms [6], or neural networks [7]. This search is justified by the Anfinsen’s “Thermodynamic Hypothesis”, claiming that a protein’s native structure is at its lowest free energy minimum [8]. The use of computational intelligence tools coupled with proteins energy approximation models (like AMBER, DISCOVER, or ECEPP/3), come from the fact that finding the exact minimum energy of a 3D structure of a protein is a very time consuming task. Furthermore, in order to tackle the complexity of the PSP problem, authors that try to predict the protein folding process use models of various resolutions. In low resolution models, atoms in the same amino acid can for instance be considered as the same entity. These low resolution models are used as the first stage of the 3D structure prediction: the backbone of the 3D conformation is determined. Then, high resolution models come next for further exploration. Such a prediction strategy is commonly used in PSP softwares like ROSETTA [9, 10] or TASSER [11].

In this paper, which is an extension of [12], we mathematically demonstrate that a particular dynamical system, used in low resolutions models to predict the backbone of the protein, is chaotic according to the Devaney’s formulation. Chaos in protein folding has been already investigated in the past years. For instance, in [13], the Lyapunov exponent of a folding process has been experimentally computed, to show that protein folding is highly complex. More precisely, the author has established that the crambin protein folding process, which is a small plant seed protein constituted by 46 amino acids from *Crambe Abyssinica*, has a positive Lyapunov exponent. In [14], an analysis of molecular dynamics simulation of a model α -helix indicates that the motion of the helix system is chaotic, *i.e.*, has nonzero Lyapunov exponents, broad-band power spectra, and strange attractors. Finally, in [15], the authors investigated the response of a protein fragment in an explicit solvent environment to very small perturbations of the atomic positions, showing that very tiny changes in initial conditions are amplified exponentially and lead to vastly different, inherently unpredictable behavior. These papers have studied experimentally the dynamics of protein

folding and stated that this process exhibits some chaotic properties, where “chaos” refers to various physical understandings of the phenomenon. They noted the complexity of the process in concrete cases, without offering a study framework making it possible to understand the origins of such a behavior.

The approach presented in this research work is different for the two following reasons. First, we focus on mathematical aspects of chaos, like the Devaney’s formulation of a chaotic dynamical system. This well-known topological notion for a chaotic behavior is one of the most established mathematical definition of unpredictability for dynamical systems. Second, we do not study the biological folding process, but the protein folding process as it is described in the 2D hydrophobic-hydrophilic (HP) lattice model [16]. In other words, we mathematically study the folding dynamics used in this model, and we wonder if this model is stable through small perturbations. For instance, what are the effects in the 2D model of changing a residue from hydrophobic to hydrophilic? Or what happens if we do not realize exactly the good rotation on the good residue, at one given stage of the 2D folding process, due to small errors in the knowledge of the protein?

Let us recall that the 2D HP square lattice model is a popular model with low resolution that focuses only on hydrophobicity by separating the amino acids into two sets: hydrophobic (H) and hydrophilic (or polar P) [17]. This model has been used several times for protein folding prediction [3, 15, 18, 19, 20]. In what follows, we show that *the folding process is unpredictable (chaotic) in the 2D HP square lattice model used for prediction*, and we investigate the consequences of this fact. Chaos here refers to our inability to make relevant prediction with this model, which does not *necessarily* imply that the biological folding dynamics is chaotic, too. In particular, we do not claim that these biological systems must try a large number of conformations in order to find the best one. Indeed, the prediction model is proven to be chaotic, but this fact is not clearly related to the impact of environmental factors on true biological protein folding.

After having established by two different proofs the chaos, as defined in the Devaney’s formulation, of the dynamical system used in the 2D model, we will deepen the evaluation of the disorder generated by this system for backbone prediction. A qualitative topological study shows that its folding dynamics is both indecomposable and unstable. Moreover, the unpredictability of the system is evaluated quantitatively too, by computing the constant of sensibility to the initial conditions and the constant of expansivity. All of these results show that the dynamical system used for backbone prediction in the 2D model has a very intense chaotic behavior and is highly unpredictable.

Consequences of these theoretical results are then outlined. More precisely, we will focus on the following questions. First, some artificial intelligence tools used for protein folding prediction are then based, for the backbone evaluation, on a dynamical system that presents several chaotic properties. It is reasonable to wonder whether these properties impact the quality of the prediction. More specifically, we will study if neural networks are able to learn a topological chaotic behavior, and if predictions resulting from this learning are close to the

reality. Moreover, the initial conformation, encompassing the sequence of amino acids, their interactions, and the effects of the outside world, are never known with infinite precision. Taking into account the fact that the model used for prediction embeds a dynamical system being sensitive to its initial condition, what can we conclude about the confidence put into the final 3D conformation? Concerning the biological aspects of the folding process, the following facts can be remarked. On the one hand, a chaotic behavior seems to be incompatible with approximately one thousand general categories of folds: this final kind of order seems in contradiction with chaos. Additionally, sensibility to initial conditions seems to be contradictory with the fact that a sequence of amino acids always folds in the same conformation, whatever the environment dependency. So, as the 2D HP lattice model for backbone prediction is chaotic whereas the whole folding process seems not, one can wonder whether this backbone prediction is founded or not. On the other hand, recent experimental researches recalled previously tend to prove that the folding process presents, at least to a certain extent, some characteristics of a chaotic behavior [13, 14, 15]. If this theory is confirmed and biological explanations are found (for instance, regulatory processes could repair or delete misfolded proteins), then this research work could appear as a first step in the theoretical study of the chaos of protein folding.

In fact, the contradiction raised above is only apparent, as it is wrong to claim that all of the sequences of amino acids always fold in a constant and well-defined conformation. More precisely, a large number of proteins, called “intrinsically unstructured proteins” or “intrinsically disordered proteins”, lay at least in part outside this rule. More than 600 proteins are proven to be of this kind: antibodies, p21 and p27 proteins, fibrinogen, casein in mammalian milk, capsid of the Tobacco mosaic virus, proteins of the capsid of bacteriophages, to name a few. Indeed, a large number of proteins have at least a disordered region of greater or lesser size. This flexibility allow them to exert various functions into an organism or to bind to various macromolecules. For instance, the p27 protein can be binded to various kind of enzymes. Furthermore, some studies have shown that between 30% and 50% of the eukaryote proteins have at least one large unstructured region [21, 22]. Hence, regular and disordered proteins can be linked to the mathematical notions of chaos as understood by Devaney, or Knudsen, which consist in the interlocking of points having a regular behavior with points whose desire is to visit the whole space.

The remainder of this paper is structured as follows. In the next section we recall some notations and terminologies on the 2D model and the Devaney’s definition of chaos. In Section 3, the folding process in the 2D model is written as a dynamical system on a relevant metrical space. Compared to [12], we have simplified the folding function and refined the metrical space to the set of all acceptable conformations. This work, which is the first contribution of this paper, has been realized by giving a complete understanding of the so-called Self-Avoiding Walk (SAW) requirement. In Sections 4 and 5, proofs of the chaotic behavior of a dynamical system used for backbone prediction, are taken from [12] and adapted to this set of acceptable conformations. This adaptation

is the second contribution of this research work. The first proof is directly achieved in Devaney’s context whereas the second one uses a previously proven result concerning chaotic iterations [23]. The following section is devoted to qualitative and quantitative evaluations of the disorder exhibited by the folding process. This is the third theoretical contribution of this extension of [12]. Consequences of this unpredictable behavior are given in Section 7. Among other things, it is regarded whether chaotic behaviors are harder to predict than “normal” behaviors or not, and if such behaviors are easy to learn. This section extends greatly the premises outlined formerly in [12]. Additionally, reasons explaining why a chaotic behavior unexpectedly leads to approximately one thousand categories of folds are proposed. This paper ends by a conclusion section, in which our contribution is summarized and intended future work is presented.

2 Basic Concepts

In the sequel S^n denotes the n^{th} term of a sequence S and V_i the i^{th} component of a vector V . The k^{th} composition of a single function f is represented by $f^k = f \circ \dots \circ f$. The set of congruence classes modulo 4 is denoted by $\mathbb{Z}/4\mathbb{Z}$. Finally, given two integers $a < b$, the following notation is used: $\llbracket a; b \rrbracket = \{a, a+1, \dots, b\}$.

2.1 2D Hydrophilic-Hydrophobic (HP) Model

HP Model

In the HP model, hydrophobic interactions are supposed to dominate protein folding. This model was formerly introduced by Dill, who considers in [17] that the protein core freeing up energy is formed by hydrophobic amino acids, whereas hydrophilic amino acids tend to move in the outer surface due to their affinity with the solvent (see Fig. 1).

In this model, a protein conformation is a “self-avoiding walk (SAW)” on a 2D or 3D lattice such that its energy E , depending on topological neighboring contacts between hydrophobic amino acids that are not contiguous in the primary structure, is minimal. In other words, for an amino-acid sequence P of length N and for the set $\mathcal{C}(P)$ of all SAW conformations of P , the chosen conformation will be $C^* = \operatorname{argmin} \{E(C)/C \in \mathcal{C}(P)\}$ [24]. In that context and for a conformation C , $E(C) = -q$ where q is equal to the number of topological hydrophobic neighbors. For example, $E(c) = -5$ in Fig. 1.

Protein Encoding

Additionally to the direct coordinate presentation, at least two other isomorphic encoding strategies for HP models are possible: relative encoding and absolute encoding. In relative encoding [1], the move direction is defined relative to the direction of the previous move. Alternatively, in absolute encoding [25],

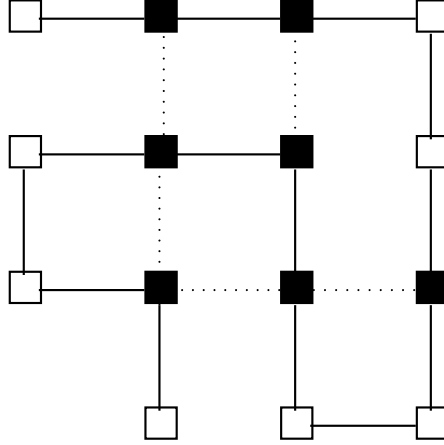


Figure 1: Hydrophilic-hydrophobic model (black squares are hydrophobic residues)

which is the encoding chosen in this paper, the direct coordinate presentation is replaced by letters or numbers representing directions with respect to the lattice structure.

For absolute encoding in the 2D square lattice, the permitted moves are: forward \rightarrow (denoted by 0), down \downarrow (1), backward \leftarrow (2), and up \uparrow (3). A 2D conformation C of $N+1$ residues for a protein P is then an element C of $\mathbb{Z}/4\mathbb{Z}^N$, with a first component equal to 0 (forward) [1]. For instance, in Fig. 1, the 2D absolute encoding is 00011123322101 (starting from the upper left corner). In that situation, at most 4^N conformations are possible when considering $N+1$ residues, even if some of them are invalid due to the SAW requirement.

2.2 Devaney's Chaotic Dynamical Systems

From a mathematical point of view, deterministic chaos has been thoroughly studied these last decades, with different research works that have provide various definitions of chaos. Among these definitions, the one given by Devaney [26] is perhaps the most well established.

Consider a topological space (\mathcal{X}, τ) and a continuous function f on \mathcal{X} . Topological transitivity occurs when, for any point, any neighborhood of its future evolution eventually overlap with any other given region. More precisely,

Definition 1 f is said to be *topologically transitive* if, for any pair of open sets $U, V \subset \mathcal{X}$, there exists $k > 0$ such that $f^k(U) \cap V \neq \emptyset$.

This property implies that a dynamical system cannot be broken into simpler subsystems. It is intrinsically complicated and cannot be simplified. Besides, a dense set of periodic points is an element of regularity that a chaotic dynamical system has to exhibit.

Definition 2 An element (a point) x is a *periodic element* (point) for f of period $n \in \mathbb{N}^*$, if $f^n(x) = x$.

Definition 3 f is said to be *regular* on (\mathcal{X}, τ) if the set of periodic points for f is dense in \mathcal{X} : for any point x in \mathcal{X} , any neighborhood of x contains at least one periodic point.

This regularity “counteracts” the effects of transitivity. Thus, due to these two properties, two points close to each other can behave in a completely different manner, leading to unpredictability for the whole system. Then,

Definition 4 (Devaney’s chaos) f is said to be *chaotic* on (\mathcal{X}, τ) if f is regular and topologically transitive.

The chaos property is related to the notion of “sensitivity”, defined on a metric space (\mathcal{X}, d) by:

Definition 5 f has *sensitive dependence on initial conditions* if there exists $\delta > 0$ such that, for any $x \in \mathcal{X}$ and any neighborhood V of x , there exist $y \in V$ and $n \geq 0$ such that $d(f^n(x), f^n(y)) > \delta$.

δ is called the *constant of sensitivity* of f .

Indeed, Banks *et al.* have proven in [27] that when f is chaotic and (\mathcal{X}, d) is a metric space, then f has the property of sensitive dependence on initial conditions (this property was formerly an element of the definition of chaos). To sum up, quoting Devaney in [26], a chaotic dynamical system “is unpredictable because of the sensitive dependence on initial conditions. It cannot be broken down or simplified into two subsystems which do not interact because of topological transitivity. And in the midst of this random behavior, we nevertheless have an element of regularity”. Fundamentally different behaviors are consequently possible and occur in an unpredictable way.

3 A Dynamical System for the 2D HP Square Lattice Model

The objective of this research work is to establish that the protein folding process, as it is described in the 2D model, has a chaotic behavior. To do so, this process must be first described as a dynamical system.

3.1 Initial Premises

Let us start with preliminaries introducing some concepts that will be useful in our approach.

The primary structure of a given protein P with $N + 1$ residues is coded by $00 \dots 0$ (N times) in absolute encoding. Its final 2D conformation has an absolute encoding equal to $0C_1^* \dots C_{N-1}^*$, where $\forall i, C_i^* \in \mathbb{Z}/4\mathbb{Z}$, is such that

$E(C^*) = \operatorname{argmin} \{E(C) / C \in \mathcal{C}(P)\}$. This final conformation depends on the repartition of hydrophilic and hydrophobic amino acids in the initial sequence.

Moreover, we suppose that, if the residue number $n+1$ is forward the residue number n in absolute encoding (\rightarrow) and if a fold occurs after n , then the forward move can only be changed into up (\uparrow) or down (\downarrow). That means, in our simplistic model, only rotations of $+\frac{\pi}{2}$ or $-\frac{\pi}{2}$ are possible.

Consequently, for a given residue that is supposed to be updated, only one of the two possibilities below can appear for its absolute move during a fold:

- $0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3$, or $3 \mapsto 0$ for a fold in the clockwise direction, or
- $1 \mapsto 0, 2 \mapsto 1, 3 \mapsto 2$, or $0 \mapsto 3$ for an anticlockwise.

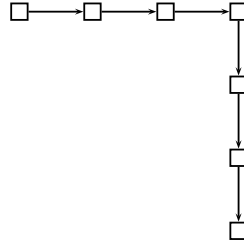
This fact leads to the following definition:

Definition 6 The *clockwise fold function* is the function $f : \mathbb{Z}/4\mathbb{Z} \rightarrow \mathbb{Z}/4\mathbb{Z}$ defined by $f(x) = x + 1(\bmod 4)$.

Obviously the dual anticlockwise fold function is $f^{-1}(x) = x - 1(\bmod 4)$.

Thus at the n^{th} folding time, a residue k is chosen and its absolute move is changed by using either f or f^{-1} . As a consequence, all of the absolute moves must be updated from the coordinate k until the last one N by using the same folding function.

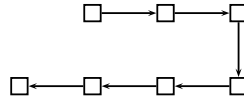
Example 1 If the current conformation is $C = 000111$, i.e.,



and if the third residue is chosen to fold by a rotation of $-\frac{\pi}{2}$ (mapping f), the new conformation will be:

$$(C_1, C_2, f(C_3), f(C_4), f(C_5), f(C_6)) = (0, 0, 1, 2, 2, 2).$$

That is,



These considerations lead to the formalization described hereafter.

3.2 Formalization and Notations

Let $N + 1$ be a fixed number of amino acids, where $N \in \mathbb{N}^*$. We define

$$\check{\mathcal{X}} = \mathbb{Z}/4\mathbb{Z}^N \times \llbracket -N; N \rrbracket^N$$

as the phase space of all possible folding processes. An element $X = (C, F)$ of this dynamical folding space is constituted by:

- A conformation of the $N+1$ residues in absolute encoding: $C = (C_1, \dots, C_N) \in \mathbb{Z}/4\mathbb{Z}^N$. Note that we do not require self-avoiding walks here.
- A sequence $F \in \llbracket -N; N \rrbracket^N$ of future folds such that, when $F_i \in \llbracket -N; N \rrbracket$ is k , it means that it occurs:
 - a fold after the k -th residue by a rotation of $-\frac{\pi}{2}$ (mapping f) at the i -th step, if $k = F_i > 0$,
 - no fold at time i if $k = 0$,
 - a fold after the $|k|$ -th residue by a rotation of $\frac{\pi}{2}$ (i.e., f^{-1}) at the i -th time, if $k < 0$.

On this phase space, the protein folding dynamic in the 2D model can be formalized as follows.

Denote by i the map that transforms a folding sequence in its first term (i.e., in the first folding operation):

$$\begin{array}{ccc} i : & \llbracket -N; N \rrbracket^N & \longrightarrow \llbracket -N; N \rrbracket \\ & F & \longmapsto F^0, \end{array}$$

by σ the shift function over $\llbracket -N; N \rrbracket^N$, that is to say,

$$\begin{array}{ccc} \sigma : & \llbracket -N; N \rrbracket^N & \longrightarrow \llbracket -N; N \rrbracket^N \\ & (F^k)_{k \in \mathbb{N}} & \longmapsto (F^{k+1})_{k \in \mathbb{N}}, \end{array}$$

and by $sign$ the function:

$$sign(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{if } x = 0, \\ -1 & \text{else.} \end{cases}$$

Remark that the shift function removes the first folding operation from the folding sequence F once it has been achieved.

Consider now the map $G : \check{\mathcal{X}} \rightarrow \check{\mathcal{X}}$ defined by:

$$G((C, F)) = (f_{i(F)}(C), \sigma(F)),$$

where $\forall k \in \llbracket -N; N \rrbracket$, $f_k : \mathbb{Z}/4\mathbb{Z}^N \rightarrow \mathbb{Z}/4\mathbb{Z}^N$ is defined by:

$$f_k(C_1, \dots, C_N) =$$

$$(C_1, \dots, C_{|k|-1}, f^{sign(k)}(C_{|k|}), \dots, f^{sign(k)}(C_N)).$$

Thus the folding process of a protein P in the 2D HP square lattice model, with initial conformation equal to $(0, 0, \dots, 0)$ in absolute encoding and a folding sequence equal to $(F^i)_{i \in \mathbb{N}}$, is defined by the following dynamical system over \check{X} :

$$\begin{cases} X^0 = ((0, 0, \dots, 0), F) \\ X^{n+1} = G(X^n), \forall n \in \mathbb{N}. \end{cases}$$

In other words, at each step n , if $X^n = (C, F)$, we take the first folding operation to realize, that is $i(F) = F^0 \in \llbracket -N; N \rrbracket$, we update the current conformation C by rotating all of the residues coming after the $|i(F)|$ -th one, which means that we replace the conformation C with $f_{i(F)}(C)$. Lastly, we remove this rotation (the first term F^0) from the folding sequence F : F becomes $\sigma(F)$.

Example 2 Let us reconsider Example 1. The unique iteration of this folding process transforms a point of \check{X} having the form $((0, 0, 0, 1, 1, 1), (+3, F^1, F^2, \dots))$ in $G(((0, 0, 0, 1, 1, 1), (+3, F^1, F^2, \dots)))$, which is equal to $((0, 0, 1, 2, 2, 2), (F^1, F^2, \dots))$.

Remark 1 Such a formalization allows the study of proteins that never stop to fold, for instance due to never-ending interactions with the environment.

Remark 2 A protein P that has finished to fold, if such a protein exists, has the form $(C, (0, 0, 0, \dots))$, where C is the final 2D structure of P . In this case, we can assimilate a folding sequence that is convergent to 0, *i.e.*, of the form $(F^0, \dots, F^n, 0, \dots)$, with the finite sequence (F^0, \dots, F^n) .

We will now introduce the SAW requirement in our formulation of the folding process in the 2D model.

3.3 The SAW Requirement

3.3.1 Towards a Basic SAW Requirement Definition

Let \mathcal{P} denotes the 2D plane and

$$\begin{aligned} p : \quad \mathbb{Z}/4\mathbb{Z}^N &\rightarrow \mathcal{P}^{N+1} \\ (C_1, \dots, C_N) &\mapsto (X_0, \dots, X_N) \end{aligned}$$

where $X_0 = (0, 0)$ and

$$X_{i+1} = \begin{cases} X_i + (1, 0) & \text{if } c_i = 0, \\ X_i + (0, -1) & \text{if } c_i = 1, \\ X_i + (-1, 0) & \text{if } c_i = 2, \\ X_i + (0, 1) & \text{if } c_i = 3. \end{cases}$$

The map p transforms an absolute encoding in its 2D representation. For instance, $p((0, 0, 0, 1, 1, 1))$ is $((0, 0); (1, 0); (2, 0); (3, 0); (3, -1); (3, -2); (3, -3))$, that is, the first figure of Example 1.

Now, for each (P_0, \dots, P_N) of \mathcal{P}^{N+1} , we denote by

$$\text{support}((P_0, \dots, P_N))$$

the set (with no repetition): $\{P_0, \dots, P_N\}$. For instance,

$$\text{support}(((0, 0); (0, 1); (0, 0); (0, 1))) = \{(0, 0); (0, 1)\}.$$

Then,

Definition 7 A conformation $(C_1, \dots, C_N) \in \mathbb{Z}/4\mathbb{Z}^N$ satisfies the *self-avoiding walk (SAW) requirement* iff the cardinality of $\text{support}(p((C_1, \dots, C_N)))$ is $N + 1$.

We can remark that Definition 7 concerns only one conformation, and not a *sequence* of conformations that occurs in a folding process.

3.3.2 Understanding the so-called SAW Requirement for a Folding Process

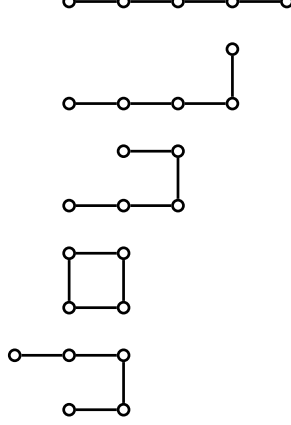
The next stage in the formalization of the protein folding process in the 2D model as a dynamical system is to take into account the self-avoiding walk (SAW) requirement, by restricting the set $\mathbb{Z}/4\mathbb{Z}^N$ of all possible conformations to one of its subsets. That is, to define precisely the set $\mathcal{C}(P)$ of acceptable conformations of a protein P having $N + 1$ residues. This stage needs a clear definition of the SAW requirement. However, as stated above, Definition 7 only focus on the SAW requirement of a given conformation, but not on a complete folding process. In our opinion, this requirement applied to the whole folding process can be understood at least in four ways.

In the first and least restrictive approach, we call it “ SAW_1 ”, we only require that the studied conformation satisfy the SAW requirement of Definition 7. It is not regarded whether this conformation is the result of a folding process that has started from $(0, 0, \dots, 0)$. Such a SAW requirement has been chosen by authors of [2] when they have proven the NP-completeness of the PSP problem.

The second approach called SAW_2 requires that, starting from the initial condition $(0, 0, \dots, 0)$, we obtain by a succession of folds a final conformation that is a self-avoiding walk. In other words, we want that the final tree corresponding to the true 2D conformation has 2 vertices with 1 edge and $N - 2$ vertices with 2 edges. For instance, the folding process of Figure 2 is acceptable in SAW_2 , even if it presents residues that overlap in an intermediate conformation. Such an approach corresponds to programs that start from the initial conformation $(0, 0, \dots, 0)$, fold it several times according to their embedding functions, and then obtain a final conformation on which the SAW property is checked: only the last conformation has to satisfy the Definition 7.

In the next approach, namely the SAW_3 requirement, it is demanded that each intermediate conformation, between the initial one and the returned (final) one, satisfy the Definition 7. It restricts the set of all conformations $\mathbb{Z}/4\mathbb{Z}^N$, for

Figure 2: Folding process acceptable in SAW_2 but not in SAW_3



a given N , to the subset \mathfrak{C}_N of conformations (C_1, \dots, C_N) such that $\exists n \in \mathbb{N}^*$, $\exists k_1, \dots, k_n \in \llbracket -N; N \rrbracket$,

$$(C_1, \dots, C_N) = G^n(((0, 0, \dots, 0), (k_1, \dots, k_n)))$$

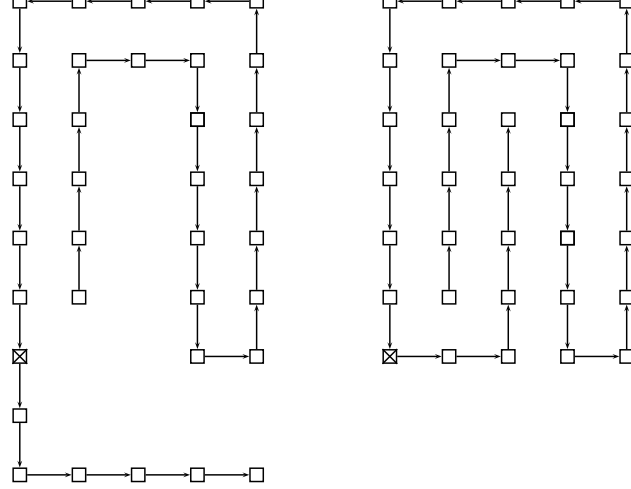
$\forall i \leq n$, the conformation $G^i(((0, \dots, 0), (k_1, \dots, k_n)))$ satisfies the Definition 7. This SAW_3 folding process requirement, which is perhaps the most usual meaning of “SAW requirement” in the literature (it is used, for instance, in [3, 15, 18, 19, 20]), has been chosen in this research work. In this approach, the acceptable conformations are obtained starting from the initial conformation $(0, 0, \dots, 0)$ and are such that all the intermediate conformations satisfy the Definition 7.

Finally, the SAW_4 approach is a SAW_3 requirement in which there is no intersection of vertex or edge during the transformation of one conformation to another. For instance, the transformation of Figure 3 is authorized in the SAW_3 approach but refused in the SAW_4 one: during the rotation around the residue identified by a cross, the structure after this residue will intersect the remainder of the “protein”. In this last approach it is impossible, for a protein folding from one plane conformation to another plane one, to use the whole space to achieve this folding.

Obviously, $SAW_4 \subsetneq SAW_3 \subseteq SAW_2 \subseteq SAW_1$. Indeed, it is easy to prove that $SAW_3 \subsetneq SAW_2$ too, but we do not know whether $SAW_2 \subsetneq SAW_1$ or not. The study of these four sets, their cardinality, characterization, and the consequence of the fact that the NP-completeness of the PSP problem has been established in SAW_1 , will be investigated in a future work.

In the present document we cannot decide what is the most reasonable approach between SAW_i , $i \in \{1, \dots, 4\}$, that is, the most close to a true natural protein folding. However, due to its complexity, the SAW_4 requirement is never used by tools that embed a 2D HP square lattice model for protein structure prediction. That is why we

Figure 3: Folding process acceptable in SAW_3 but not in SAW_4



will consider, in this research work, that the so-called “SAW requirement” for a 2D folding process corresponds to the SAW_3 approach detailed previously. Indeed, it is the most used one, and we only want to study the ability of PSP software to find the most probable 2D conformation. Thus, in what follows, the set of acceptable conformations with $N + 1$ residues will be the set \mathfrak{C}_N (*i.e.*, $\mathcal{C}(P) = \mathfrak{C}_N$).

3.4 A Metric for the Folding Process

We define a metric d over $\mathcal{X} = \mathfrak{S}_N \times \llbracket -N; N \rrbracket^N$ by:

$$d(X, \tilde{X}) = d_C(C, \check{C}) + d_F(F, \check{F}).$$

where

$$\begin{cases} \delta(a, b) = 0 \text{ if } a = b, \text{ otherwise } \delta(a, b) = 1, \\ d_C(C, \check{C}) = \sum_{k=1}^N \delta(C_k, \check{C}_k) 2^{N-k}, \\ d_F(F, \check{F}) = \frac{9}{2N} \sum_{k=0}^{\infty} \frac{|F^k - \check{F}^k|}{10^{k+1}}. \end{cases}$$

This new distance for the dynamical description of the protein folding process in the 2D HP square lattice model can be justified as follows. The integral part of the distance between two points $X = (C, F)$ and $\tilde{X} = (\check{C}, \check{F})$ of \mathcal{X} measures the differences between the current 2D conformations of X and \tilde{X} . More precisely, if $d_C(C, \check{C})$ is in $\llbracket 2^{N-(k+1)}; 2^{N-k} \rrbracket$, then the first k terms in the acceptable conformations C and \check{C} (their absolute encodings) are equal, whereas the $k + 1^{th}$ terms differ: their 2D conformations will differ after the

$k+1$ -th residue. If the decimal part of $d(X, \tilde{X})$ is between $10^{-(k+1)}$ and 10^{-k} , then the next k foldings of C and \tilde{C} will occur in the same place (residue), same order, and same angle. The decimal part of $d(X, \tilde{X})$ will then decrease as the duration where the folding process is similar increases.

More precisely, $F^k = \tilde{F}^k$ (same residue and same angle of rotation at the k -th stage of the 2D folding process) if and only if the $k+1^{th}$ digit of this decimal part is 0. Lastly, $\frac{9}{2^N}$ is just a normalization factor.

For instance, if we know where are now the $N+1$ residues of our protein P in the lattice (knowledge of the correct conformation), and if we have discovered what will be its k next foldings, then we know that the point $X = (C, F)$ describing the folding process of the considered protein in the 2D model, will be “somewhere” into the ball $\mathcal{B}(C, 10^{-k})$, that is, very close to the point (C, F) if k is large.

Example 3 Let us consider two points

- $X = ((0, 0, 0, 1, 1, 1), (3, -4, 2))$,
- and $X' = ((0, 0, 0, 1, 1, 1), (3, -4, -6))$

of \mathcal{X} . We note $X = (C, F)$ and $X' = (C', F')$. $d_C(C, C') = 0$, then these two points have the same current (first) conformation. As $d_F(F, F') = \frac{9}{2 \times 6} \frac{|2 - (-6)|}{10^3} = 0.006$ is in $[10^{-3}; 10^{-2}]$, we can deduce that the two next foldings of X and of X' will lead to identical conformations, whereas the third folding operation will lead to different conformations. A possible way to represent these two points of the phase space is to draw the successive conformations induced by these points, as illustrated in Figure 4.

Example 4 Figure 5 contains the representation of the two “points” $X = ((0, 0, 0, 1, 1, 1), (3, -4, 2))$ and $X' = ((0, 0, 1, 2, 2, 2), (-4, -5))$. Let $X = (C, F)$ and $X' = (C', F')$. We have

$$d_C(C, C') = 2^{6-3} + 2^{6-4} + 2^{6-5} + 2^{6-6} = 15$$

and $d_F = \frac{9}{12} \left(\frac{|3 - (-4)|}{10} + \frac{|-4 - (-5)|}{100} + \frac{|2 - 0|}{1000} \right) = 0.534$, then $d(X, X') = 15.534$.

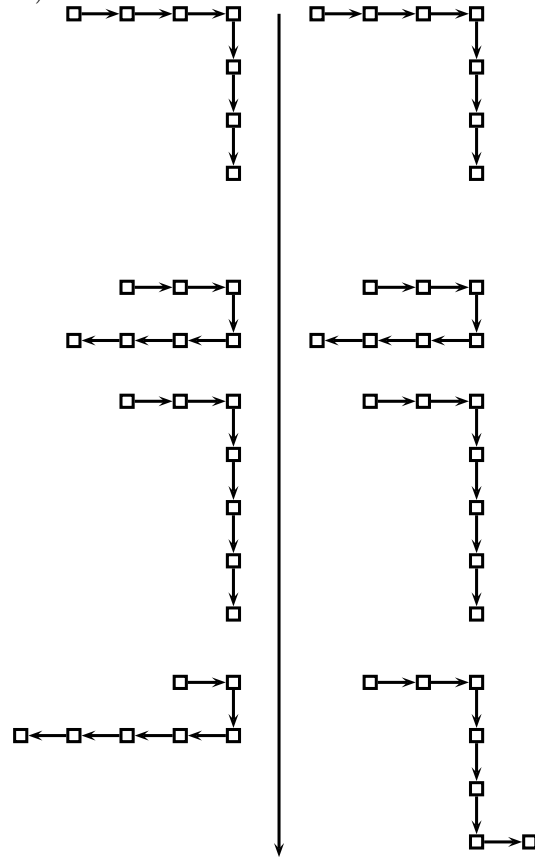
As 15 is in $[2^3; 2^4]$, we can conclude that the absolute encodings of the two initial conformations are similar for the first $k = N - 4 = 2$ terms.

4 Folding Process in 2D Model is Chaotic

4.1 Motivations

In our topological description of the protein folding process in the 2D model, all the information is embedded into the folding sequence F . Indeed, roughly speaking, it is as if nature has a function \mathcal{N} that translates a protein P having a linear conformation $(0, \dots, 0)$ into an environment E , in a folding sequence F , *i.e.*,

Figure 4: Representation of $X = ((0, 0, 0, 1, 1, 1), (3, -4, 2))$ and $X' = ((0, 0, 0, 1, 1, 1), (3, -4, -6))$ of the phase space \mathcal{X} (X is in left part of the figure, X' is its right part).



$F = \mathcal{N}(P, E)$. Having this “natural” folding sequence F , we are able to obtain its true conformation in the 2D model, by computing $G^n(((0, \dots, 0), F))$, where n is the size of F . On our side, we have only a partial knowledge of the environment E and of the protein P (exact interactions between atoms). We thus consider \tilde{P} and \tilde{E} , as close as we can to P and E respectively. Moreover, we have only a model $\tilde{\mathcal{N}}$ of \mathcal{N} as, for instance, we use various approximations: models for free energy, approximations of hydrophobic/hydrophilic areas and electro-polarity, etc. This is why we can only deduce an approximation $\tilde{F} = \tilde{\mathcal{N}}(\tilde{P}, \tilde{E})$ of the natural folding sequence $F = \mathcal{N}(P, E)$. One important motivation of this work is to determine whether, having an approximation \tilde{F} of F , we obtain a final conformation $\tilde{C} = G^{\tilde{n}}(((0, \dots, 0), \tilde{F}))_0$ close to the natural conformation $C = G^n(((0, \dots, 0), F))_0$ or not. In this last sentence, n and \tilde{n} are the sizes of F and \tilde{F} respectively, and the terms “approximation” and “close” can be understood by using d_F and d_C , respectively. To sum up, even if we cannot have access with an infinite precision to all of the forces that participate to the folding process, *i.e.*, even if we only know an approximation $X'^0 = ((0, \dots, 0), \tilde{F})$ of $X^0 = ((0, \dots, 0), F)$, can we claim that the predicted conformation $X'^{n_1} = G^{n_1}(((0, \dots, 0), \tilde{F}))$ still remains close to the true conformation $X^{n_2} = G^{n_2}(((0, \dots, 0), F))$? Or, on the contrary, do we have a chaotic behavior, a kind of butterfly effect that magnifies any error on the evaluation of the forces in presence?

Raising such a question leads to the study of the dynamical behavior of the folding process.

4.2 Continuity of the Folding Process

We will now give a first proof of the chaotic behavior of the protein folding dynamics in the 2D model. To do so, we must establish first that G is a continuous map on (\mathcal{X}, d) . Indeed, the mathematical theory of chaos only studies dynamical systems defined by a recurrence relation of the form $X^{n+1} = G(X^n)$, with G continuous.

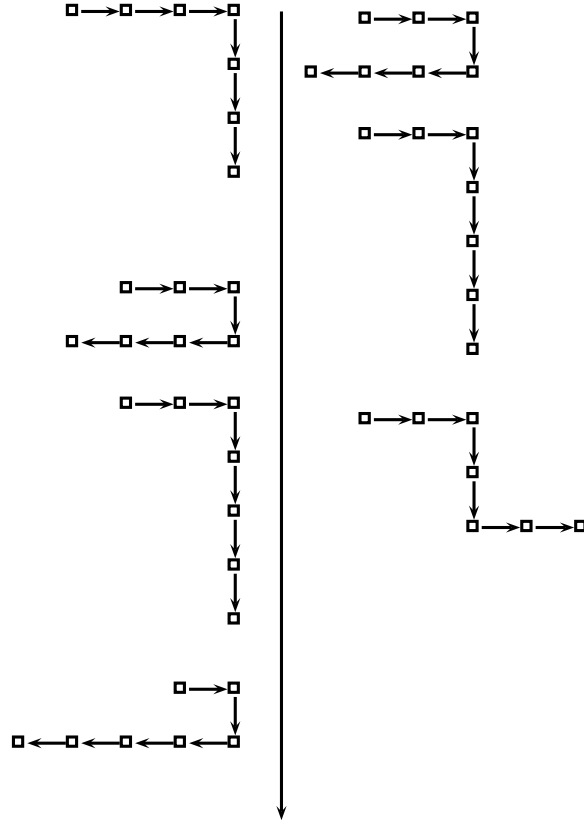
Proposition 1 *G is a continuous map on (\mathcal{X}, d) .*

PROOF We will use the sequential characterization of the continuity. Let $(X^n)_{n \in \mathbb{N}} = ((C^n, F^n))_{n \in \mathbb{N}} \in \mathcal{X}^{\mathbb{N}}$, such that $X^n \rightarrow X = (\tilde{C}, \tilde{F})$. We will then show that $G(X^n) \rightarrow G(X)$. Let us remark that $\forall n \in \mathbb{N}$, F^n is a sequence: F is thus a sequence of sequences.

On the one hand, as $X^n = (C^n, F^n) \rightarrow (\tilde{C}, \tilde{F})$, we have $d_C(C^n, \tilde{C}) \rightarrow 0$, thus $\exists n_0 \in \mathbb{N}$, $n \geq n_0 \Rightarrow d_C(C^n, \tilde{C}) = 0$. That is, $\forall n \geq n_0$ and $\forall k \in \llbracket 1; \mathbb{N} \rrbracket$, $\delta(C_k^n, \tilde{C}_k) = 0$, and so $C^n = \tilde{C}$, $\forall n \geq n_0$. Additionally, since $d_F(F^n, \tilde{F}) \rightarrow 0$, $\exists n_1 \in \mathbb{N}$ such that we have $d_F(F_1^n, \tilde{F}) \leq \frac{1}{10}$. As a consequence, $\exists n_1 \in \mathbb{N}$, $\forall n \geq n_1$, the first term of the sequence F^n is \tilde{F}^0 : $i(F^n) = i(\tilde{F})$. So, $\forall n \geq \max(n_0, n_1)$, $f_{i(F^n)}(C^n) = f_{i(\tilde{F})}(\tilde{C})$, and then $f_{i(F^n)}(C^n) \rightarrow f_{i(\tilde{F})}(\tilde{C})$.

On the other hand, $\sigma(F^n) \rightarrow \sigma(\tilde{F})$. Indeed, $F^n \rightarrow \tilde{F}$ implies $\sum_{k=0}^{\infty} \frac{|(F^n)^k - \tilde{F}^k|}{10^{k+1}} \rightarrow$

Figure 5: Representation of $X = ((0, 0, 0, 1, 1, 1), (3, -4, 2))$ and $X' = ((0, 0, 1, 2, 2, 2), (-4, -5))$ of the phase space \mathcal{X} (X is in left part of the figure, X' is its right part).



0, from which we obtain $\frac{1}{10} \sum_{k=0}^{\infty} \frac{|(F^n)^{k+1} - \check{F}^{k+1}|}{10^{k+1}} \rightarrow 0$, so $\sum_{k=0}^{\infty} \frac{|\sigma(F^n)^k - \sigma(\check{F})^k|}{10^{k+1}}$ converges towards 0. Finally, $\sigma(F^n) \rightarrow \sigma(\check{F})$.

Since we have shown that $f_{i(F^n)}(C^n) \rightarrow f_{i(\check{F})}(\check{C})$ and $\sigma(F^n) \rightarrow \sigma(\check{F})$, we conclude that $G(X^n) \rightarrow G(X)$.

It is now possible to study the chaotic behavior of the folding process.

4.3 A First Fundamental Lemma

Let us start by introducing the following fundamental lemma, meaning that we can transform any acceptable conformation to any other one in SAW_3 , by finding a relevant folding sequence.

Lemma 1 $\forall C, C'$ in $\mathfrak{C}_{\mathbb{N}}$, $\exists n \in \mathbb{N}^*$ and k_1, \dots, k_n in $\llbracket -\mathbb{N}; \mathbb{N} \rrbracket$ s.t.

$$G^n((C, (k_1, \dots, k_n, 0, \dots))) = (C', (0, \dots, 0)).$$

PROOF As we consider conformations of $\mathfrak{C}_{\mathbb{N}}$, we take place in the SAW_3 requirement, and thus there exist $n_1, n_2 \in \mathbb{N}^*$ and $l_1, \dots, l_{n_1}, m_1, \dots, m_{n_2}$ in $\llbracket -\mathbb{N}; \mathbb{N} \rrbracket$ such that $C = G^{n_1}(((0, \dots, 0), (l_1, \dots, l_{n_1})))$ and $C' = G^{n_2}(((0, \dots, 0), (m_1, \dots, m_{n_2})))$. The result of the lemma is then obtained with

$$(k_1, \dots, k_n) = (-l_{n_1}, -l_{n_1-1}, \dots, -l_1, m_1, \dots, m_{n_2}).$$

4.4 Regularity and Transitivity

Let us recall that the first component X_0 of $X = (C, F)$ is the current conformation C of the protein and the second component X_1 is its future folding process F . We will now prove that,

Proposition 2 *Folding process in 2D model is regular.*

PROOF Let $X = (C, F) \in \mathcal{X}$ and $\varepsilon > 0$. Then we define $k_0 = -\lfloor \log_{10}(\varepsilon) \rfloor$ and \tilde{X} such that:

1. $\tilde{X}_0 = C$,
2. $\forall k \leq k_0, G^k(\tilde{X})_1 = G^k(X)_1$,
3. $\forall i \in \llbracket 1; n \rrbracket, G^{k_0+i}(\tilde{X})_1 = k_i$,
4. $\forall i \in \mathbb{N}, G^{k_0+n+i+1}(\tilde{X})_1 = G^i(\tilde{X})_1$,

where k_1, \dots, k_n are integers given by Lemma 1 with $C = G^{k_0}(X)_0$ and $C' = X_0$. Such an \tilde{X} is a periodic point for G into the ball $\mathcal{B}(X, \varepsilon)$. (1) and (2) are to make \tilde{X} ε -close to X , (3) is for mapping the conformation $G^{k_0}(\tilde{X})_0$ into C in at most n foldings. Lastly, (4) is for the periodicity of the folding process.

Let us now consider the second property required in the Devaney's definition. Instead of proving the transitivity of G , we will establish its strong transitivity:

Definition 8 A dynamical system (\mathcal{X}, f) is strongly transitive if $\forall x, y \in \mathcal{X}$, $\forall r > 0$, $\exists z \in \mathcal{X}$, $d(z, x) \leq r \Rightarrow \exists n \in \mathbb{N}^*$, $f^n(z) = y$.

In other words, for all $x, y \in \mathcal{X}$, it is possible to find a point z in the neighborhood of x such that an iterate $f^n(z)$ is y . Obviously, strong transitivity implies transitivity. Let us now prove that,

Proposition 3 *Folding process in 2D model is strongly transitive.*

PROOF Let $X_A = (C_A, F_A)$, $X_B = (C_B, F_B)$, and $\varepsilon > 0$. We will show that $X \in \mathcal{B}(X_A, \varepsilon)$ and $n \in \mathbb{N}$ can be found such that $G^n(X) = X_B$. Let $k_0 = -\lfloor \log_{10}(\varepsilon) \rfloor$ and $\tilde{X} = G^{k_0}(C_A, F_A)$ denoted by $\tilde{X} = (\tilde{C}, \tilde{F})$. According to Lemma 1 applied to \tilde{C} and C_B , $\exists k_1, \dots, k_n$ in $\llbracket -N, N \rrbracket$ such that

$$G^n((\tilde{C}, (k_1, \dots, k_n, 0, \dots))) = (C_B, (0, \dots)).$$

Let us define $X = (C, F)$ in the following way:

1. $C = C_A$,
2. $\forall k \leq k_0, F^k = F_A^k$,
3. $\forall i \in \llbracket 1; n \rrbracket, F^{k_0+i} = k_i$,
4. $\forall i \in \mathbb{N}, F^{k_0+n+i+1} = F_B^i$.

This point X is thus an element of $\mathcal{B}(X_A, \varepsilon)$ (due to 1, 2) being such that $G^{k_0+n+1}(X) = X_B$ (by using 3, 4). As a consequence, G is strongly transitive.

Strong transitivity states that being as close as possible of the true folding process (2D model) is not a guarantee of success. Indeed, let P be a protein under interest and F its natural folding process in the 2D model. Then, for all possible conformation C of the square lattice, there exists a folding sequence \tilde{F} very close to F leading to C . More precisely, for any $\varepsilon > 0$ (as small as possible), an infinite number of folding sequences are in $\mathcal{B}_{d_F}(F, \varepsilon)$ and lead to C . The strong transitivity property implies that without the knowledge of the *exact* initial condition (the natural folding process, and thus the exact free energy), all the conformations are possible. Additionally, no conformation of the square lattice can be discarded when studying a protein folding in the 2D HP square lattice model: the dynamical system obtained by such a formalization is intrinsically complicated and cannot be decomposed or simplified. Furthermore, this trend to visit the whole space of acceptable conformations is counteracted by elements of regularity stated before: it is even impossible to dress a kind of qualitative description of the dynamics in the 2D square lattice model, as two points close to each other can have fundamentally different behaviors.

4.5 Chaotic behavior of the folding process

As G is regular and (strongly) transitive, we have:

Theorem 1 *The folding process G in the 2D model is chaotic according to Devaney.*

Consequently this process is highly sensitive to its initial conditions. If the 2D model can accurately describe the natural process, then Theorem 1 implies that even a minute difference on an intermediate conformation of the protein, in forces that act in the folding process, or in the position of an atom, can lead to enormous differences in its final conformation, even over fairly small timescales. This is the so-called butterfly effect. In particular, it seems very difficult to predict, in this 2D model, the structure of a given protein by using the knowledge of the structure of similar proteins. Let us remark that the whole 3D folding process with real torsion angles is obviously more complex than this 2D HP model. And finally, that chaos refers to our incapacity to make good prediction, it does not mean that the biological process is a random one.

Before studying some practical aspects of this unpredictability in Section 7, we will initiate a second proof of the chaotic behavior of this process and deepen its chaotic properties.

5 Outlines of a second proof

In this section a second proof of the chaotic behavior of the protein folding process is given. It is proven that the folding dynamics can be modeled as chaotic iterations (CIs). CIs are a tool used in distributed computing and in the computer science security field [28] that has been established to be chaotic according to Devaney [29].

This second proof is the occasion to introduce these CIs, which will be used at the end of this paper to study whether a chaotic behavior is really more difficult to learn with a neural network than a “normal” behavior.

5.1 Chaotic Iterations: Recalls of Basis

Let us consider a *system* with a finite number $N \in \mathbb{N}^*$ of elements (or *cells*), so that each cell has a Boolean *state*. A sequence of length N of Boolean states of the cells corresponds to a particular *state of the system*. A sequence, whose elements are subsets of $\llbracket 1; N \rrbracket$, is called a *strategy*. The set of all strategies is denoted by \mathbb{S} and the set \mathbb{B} is for the Booleans $\{0, 1\}$.

Definition 9 Let $f : \mathbb{B}^N \rightarrow \mathbb{B}^N$ be a function and $S \in \mathbb{S}$ be a strategy. The so-called *chaotic iterations* (CIs) are defined by $x^0 \in \mathbb{B}^N$ and $\forall n \in \mathbb{N}^*, \forall i \in \llbracket 1; N \rrbracket$,

$$x_i^n = \begin{cases} x_i^{n-1} & \text{if } i \notin S^n \\ (f(x^{n-1}))_i & \text{if } i \in S^n. \end{cases}$$

In other words, at the n^{th} iteration, only the S^n -th cells are “iterated”. Let us notice that the term “chaotic”, in the name of these iterations, has *a priori* no link with the mathematical theory of chaos recalled previously. We will now recall that CIs can be written as a dynamical system, and characterize functions f such that their CIs are chaotic according to Devaney [23].

5.2 CIs and Devaney’s chaos

Let $f : \mathbb{B}^N \longrightarrow \mathbb{B}^N$. We define $F_f : \llbracket 1; N \rrbracket \times \mathbb{B}^N \longrightarrow \mathbb{B}^N$ by:

$$F_f(k, E) = \left(E_j \cdot \delta(k, j) + f(E)_k \cdot \overline{\delta(k, j)} \right)_{j \in \llbracket 1; N \rrbracket},$$

where $+$ and \cdot are the Boolean addition and product operations, and \bar{x} is for the negation of x .

We have proven in [23] that chaotic iterations can be described by the following dynamical system:

$$\begin{cases} X^0 \in \tilde{\mathcal{X}} \\ X^{k+1} = \tilde{G}_f(X^k), \end{cases}$$

where $\tilde{G}_f((S, E)) = (\sigma(S), F_f(i(S), E))$, and $\tilde{\mathcal{X}}$ is a metric space for an ad hoc distance such that \tilde{G} is continuous on \mathcal{X} [23].

Let us now consider the following oriented graph, called *graph of iterations*. Its vertices are the elements of \mathbb{B}^N , and there is an arc from $x = (x_1, \dots, x_i, \dots, x_N) \in \mathbb{B}^N$ to $x = (x_1, \dots, \bar{x}_i, \dots, x_N)$ if and only if $F_f(i, x) = (x_1, \dots, \bar{x}_i, \dots, x_N)$. If so, the label of the arc is i . In the following, this graph of iterations will be denoted by $\Gamma(f)$.

We have proven in [30] that:

Theorem 2 *Functions $f : \mathbb{B}^N \rightarrow \mathbb{B}^N$ such that \tilde{G}_f is chaotic according to Devaney, are functions such that the graph $\Gamma(f)$ is strongly connected.*

We will now show that the protein folding process can be modeled as chaotic iterations, and conclude the proof by using the theorem recalled above.

5.3 Protein Folding as Chaotic Iterations

The attempt to use chaotic iterations in order to model protein folding can be justified as follows. At each iteration, the same process is applied to the system (*i.e.*, to the conformation), that is the folding operation. Additionally, it is not a necessity that all of the residues fold at each iteration: indeed it is possible that, at a given iteration, only some of these residues folds. Such iterations, where not all the cells of the considered system are to be updated, are exactly the iterations modeled by CIs.

Indeed, the protein folding process with folding sequence $(F^n)_{n \in \mathbb{N}}$ consists in the following chaotic iterations: $C^0 = (0, 0, \dots, 0)$ and,

$$C_{|i|}^{n+1} = \begin{cases} C_{|i|}^n & \text{if } i \notin S^n \\ f^{sign(i)}(C^n)_i & \text{else} \end{cases},$$

where the chaotic strategy is defined by $\forall n \in \mathbb{N}$, $S^n = \llbracket -\mathbf{N}; \mathbf{N} \rrbracket \setminus \llbracket -F^n; F^n \rrbracket$.

Thus, to prove that the protein folding process is chaotic as defined by Devaney, is equivalent to prove that the graph of iterations of the CIs defined above is strongly connected. This last fact is obvious, as it is always possible to find a folding process that map any conformation $(C_1, \dots, C_N) \in \mathfrak{C}_N$ to any other $(C'_1, \dots, C'_N) \in \mathfrak{C}_N$ (this is Lemma 1).

Let us finally remark that it is easy to study processes s.t. more than one fold occur per time unit, by using CIs. This point will be deepened in a future work. We will now investigate some consequences resulting from the chaotic behavior of the folding process.

6 Qualitative and quantitative evaluations

Behaviors qualified as “chaos” are too complicated to be encompassed by only one rigorous definition, as perfect as it could be. Indeed, the mathematical theory of chaos brings several nonequivalent definitions for a complex, unpredictable dynamical system, each of them highlighting this complexity in a well-defined but restricted understanding. This is why, in this section, we continue the evaluation of the chaotic behavior of the 2D folding dynamical system initiated by the proof of the Devaney’s chaos.

6.1 Qualitative study

First of all, the transitivity property implies the indecomposability of the system:

Definition 10 A dynamical system (\mathcal{X}, f) is indecomposable if it is not the union of two closed sets $A, B \subset \mathcal{X}$ such that $f(A) \subset A, f(B) \subset B$.

Thus it is impossible to reduce, in the 2D model, the set of protein foldings in order to simplify its complexity. Furthermore, the folding process has the instability property:

Definition 11 A dynamical system (\mathcal{X}, f) is unstable if for all $x \in \mathcal{X}$, the orbit $\gamma_x : n \in \mathbb{N} \mapsto f^n(x)$ is unstable, that is: $\exists \varepsilon > 0, \forall \delta > 0, \exists y \in \mathcal{X}, \exists n \in \mathbb{N}, d(x, y) < \delta$ and $d(\gamma_x(n), \gamma_y(n)) \geq \varepsilon$.

This property, which is implied by the sensitive dependence to the initial conditions, leads to the fact that in all of the neighborhoods of any x , there are points that are separated from x under iterations of f . We thus can claim that the behavior of the folding process is unstable.

6.2 Quantitative measures

One of the most famous measures in the theory of chaos is the constant of sensitivity given in Definition 5. Intuitively, a function f having a constant of sensitivity equal to δ implies that there exists points arbitrarily close to any point

x that *eventually* separate from x by at least δ under some iterations of f . This induces that an arbitrarily small error on an initial condition *may* be magnified upon iterations of f . The sensitive dependence on the initial conditions is a consequence of regularity and transitivity in a metrical space [27]. However, the constant of sensitivity δ can be obtained by proving the property without using Banks' theorem.

Proposition 4 *Folding process in the 2D model has sensitive dependence on initial conditions on (\mathcal{X}, d) and its constant of sensitivity is at least equal to 2^{N-1} .*

PROOF Let $X = (C, F) \in \mathcal{X}$, $r > 0$, $B = \mathcal{B}(X, r)$ an open ball centered in X , and $k_0 \in \mathbb{Z}$ such that $10^{-k_0-1} \leq r < 10^{-k_0}$. We define \tilde{X} by:

- $\tilde{C} = C$,
- $\tilde{F}^k = F^k$, $\forall k \in \mathbb{N}$ such that $k \leq k_0$,
- $\tilde{F}^{k_0+1} = 1$ if $|F^{k_0+1}| \neq 1$, and $\tilde{F}^{k_0+1} = -F^{k_0+1}$ else.
- $\forall k \geq k_0 + 2$, $\tilde{F}^k = -F^k$.

Only two cases can occur:

1. If $|F^{k_0+1}| \neq 1$, then

$$\begin{aligned}
& d\left(G^{k_0+1}(X), G^{k_0+1}(\tilde{X})\right) \\
&= 2^{N-1} + 2^{N-F^{k_0+1}} + \frac{9}{2N} \sum_{k=k_0+1}^{\infty} \frac{|F^k - \tilde{F}^k|}{10^{k+1}} \\
&= 2^{N-1} + 2^{N-F^{k_0+1}} + \frac{9}{2N} \sum_{k=k_0+1}^{\infty} \frac{2N}{10^{k+1}} \\
&= 2^{N-1} + 2^{N-F^{k_0+1}} + 9 \frac{1}{10^{k_0+2}} \frac{1}{1 - \frac{1}{10}} \\
&= 2^{N-1} + 2^{N-F^{k_0+1}} + \frac{1}{10^{k_0+1}}.
\end{aligned}$$

2. Else, $d\left(G^{k_0+1}(X), G^{k_0+1}(\tilde{X})\right) = 2^{N-1} + \frac{1}{10^{k_0+1}}$.

In all of these cases, the sensibility to the initial condition is greater than 2^{N-1} .

Let us now recall another common quantitative measure of disorder of a dynamical system.

Definition 12 A function f is said to have the property of *expansivity* if

$$\exists \varepsilon > 0, \forall x \neq y, \exists n \in \mathbb{N}, d(f^n(x), f^n(y)) \geq \varepsilon.$$

Then ε is the *constant of expansivity* of f : an arbitrarily small error on any initial condition is *always* amplified until ε .

Proposition 5 *The folding process in the 2D model is an expansive chaotic system on (\mathcal{X}, d) . Its constant of expansivity is at least equal to 1.*

PROOF Let $X = (C, F)$ and $X' = (C', F')$ such that $X \neq X'$.

- If $C \neq C'$, then $\exists k_0 \in \llbracket 1; \mathbb{N} \rrbracket, C_{k_0} \neq C'_{k_0}$. So,

$$d(G^0(X), G^0(X')) \geq 2^{\mathbb{N}-k_0} \geq 1.$$

- Else $F' \neq F$. Let $k_0 = \min \{k \in \mathbb{N}, F^k \neq F'^k\}$. Then $\forall k < k_0, G^k(X) = G^k(X')$. Let $\check{X} = (\check{C}, \check{F}) = G^{k_0-1}(X) = G^{k_0-1}(X')$.

Then $d(G^{k_0}(X), G^{k_0}(X'))$

$$\begin{aligned} &\geq d_C(f_{F^{k_0}}(\check{C}_1, \dots, \check{C}_{\mathbb{N}}), f_{F'^{k_0}}(\check{C}_1, \dots, \check{C}_{\mathbb{N}})) \\ &\geq d_C\left(\left(\check{C}_1, \dots, \check{C}_{|F^{k_0}|-1}, f^{sign(F^{k_0})}(\check{C}_{|F^{k_0}|})\right), \dots, \right. \\ &\quad \left. f^{sign(F^{k_0})}(\check{C}_{\mathbb{N}})\right), \left(\check{C}_1, \dots, \check{C}_{|F'^{k_0}|-1}, \right. \\ &\quad \left. f^{sign(F'^{k_0})}(\check{C}_{|F'^{k_0}|})\right), \dots, f^{sign(F'^{k_0})}(\check{C}_{\mathbb{N}})\bigg) \\ &\geq 2^{\mathbb{N}-\max(|F^{k_0}|, |F'^{k_0}|)} \\ &\geq 1. \end{aligned}$$

So the result is established.

7 Consequences

7.1 Is Chaotic Behavior Incompatible with Approximately one Thousand Folds?

Results established previously only concern the folding process in the 2D HP square lattice model. At this point, it is natural to wonder if such a model, being a reasonable approximation of the true natural process, is chaotic because this natural process is chaotic too. Indeed, claiming that the natural protein folding process is chaotic seems to be contradictory with the fact that only approximately one thousand folds have been discovered this last decade [31]. The number of proteins that have an understood 3D structure increases largely year after year. However the number of new categories of folds seems to be limited by a fixed value approximately equal to one thousand. Indeed, there is no contradiction as a chaotic behavior does not forbid a certain form of order.

As stated before, chaos only refers to limitations in prediction. For example, seasons are not forbidden even if weather forecast has a non-intense chaotic behavior. A similar regularity appears in brains: even if hazard and chaos play an important role on a microscopic scale, a statistical order appears in the neural network.

That is, a certain order can emerge from a chaotic behavior, even if it is not a rule of thumb. More precisely, in our opinion these thousand folds can be related to basins of attractions or strange attractors of the dynamical system, objects that are well described by the mathematical theory of chaos. Thus, it should be possible to determine all of the folds that can occur, by refining our model and looking for its basins of attractions with topological tools. However, this assumption still remains to be investigated.

7.2 Is Artificial Intelligence able to Predict Chaotic Dynamic?

We will now focus on the impact of using a chaotic model for prediction. We give some results on two kinds of experiments, both using neural networks. Firstly, we will study whether a (mathematical) chaotic behavior can be learned by a neural network or not. Therefore, we design a global recurrent network that models the function F_f introduced in the previous section and we show that it is more difficult to train the network when f is chaotic. These considerations have been formerly proposed in [32] and are extended here. Secondly, we will try to learn the future conformation of proteins that consist of a small number of residues. Our objective is to assess if a neural network can learn the future conformation given the current one and a sequence of a few folds.

In this work, we choose to train a classical neural network architecture: the MultiLayer Perceptron, a model of network widely used and well-known for its universal approximation property [33]. Let us notice that for the first kind of experiments global feedback connections are added, in order to have a proper modeling of chaotical iterations, while for the latter kind of experiments the MLPs used are feed-forward ones. In both cases we consider networks having sigmoidal hidden neurons and output neurons with a linear activation function. They are trained using the Limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm with Wolfe linear search. The training process can either be controlled by the number of network parameters (weights and biases) updates, also called epochs, or by a mean square error criterion.

7.2.1 Can a Neural Network Learn Chaotic Functions?

Experimental Protocol

We consider $f : \mathbb{B}^N \rightarrow \mathbb{N}^N$, strategies of singletons ($\forall n \in \mathbb{N}, S^n \in \llbracket 1; \mathbb{N} \rrbracket$), and a MLP that recognize F_f . That means, for all $(k, x) \in \llbracket 1; \mathbb{N} \rrbracket \times \mathbb{B}^N$, the response of the output layer to the input (k, x) is $F_f(k, x)$. We thus connect the output layer to the input one as it is depicted in Fig. 6, leading to a global recurrent artificial neural network working as follows [32].

At the initialization stage, the network receives a Boolean vector $x^0 \in \mathbb{B}^N$ as input state, and $S^0 \in \llbracket 1; N \rrbracket$ in its input integer channel $i()$. Thus, $x^1 = F_f(S^0, x^0) \in \mathbb{B}^N$ is computed by the neural network. This state x^1 is published as an output. Additionally, x^1 is sent back to the input layer, to act as Boolean state in the next iteration. Finally, at iteration number n , the recurrent neural network receives the state $x^n \in \mathbb{B}^N$ from its output layer and $i(S^n) \in \llbracket 1; N \rrbracket$ from its input integer channel $i()$. It can thus calculate $x^{n+1} = F_f(i(S^n), x^n) \in \mathbb{B}^N$, which will be the new output of the network. Obviously, this particular MLP produces exactly the same values as CIs with update function f . That is, such MLPs are equivalent, when working with $i(s)$, to CIs with f as update function and strategy S [32].

Let us now introduce the two following functions:

- $f_1(x_1, x_2, x_3) = (\overline{x_1}, \overline{x_2}, \overline{x_3})$,
- $f_2(x_1, x_2, x_3) = (\overline{x_1}, x_1, x_2)$.

It can easily be checked that these functions satisfy the hypothesis of Theorem 2, thus their CIs are chaotic according to Devaney. Then, when the MLP defined previously learns to recognize F_{f_1} or F_{f_2} , it tries to learn these CIs, that is, a chaotic behavior as defined by Devaney [32]. On the contrary, the function

$$g(x_1, x_2, x_3) = (\overline{x_1}, x_2, x_3)$$

is such that $\Gamma(g)$ is not strongly connected. In this case, due to Theorem 2, the MLP does not learn a chaotic process. We will now recall the study of the training process of functions F_{f_1} , F_{f_2} , and F_g [32], that is to say, the ability to learn one iteration of CIs.

Experimental Results

For each neural network we have considered MLP architectures with one and two hidden layers, with in the first case different numbers of hidden neurons. Thus we will have different versions of a neural network modeling the same iteration function [32]. Only the size and number of hidden layers may change, since the numbers of inputs and output neurons are fully specified by the function. The training is performed until the learning error is lower than a chosen threshold value (10^{-2}).

Table 1 gives for each considered neural network the mean number of epochs needed to learn one iteration in their ICs, and a success rate that reflects a successful training in less than 1000 epochs. Both values are computed considering 25 trainings with random weights and biases initialization. These results highlight several points [32]. First, the two hidden layer structure seems to be quite inadequate to learn chaotic behaviors. Second, training networks so that they behave chaotically seems to be difficult for these simplistic functions only iterated one time, since they need in average more epochs to be correctly trained. In the case of the two hidden layer network topology, a comparison of the mean number of epochs needed for a successful learning of 10 chaotic functions with that obtained for 10 non chaotic functions reinforces the previous observation.

Table 1: Results of some iteration functions learning, using different recurrent MLP architectures

Function	One hidden layer			
	8 neurons		10 neurons	
	Mean epoch	Success rate	Mean epoch	Success rate
f_1	82.21	100%	73.44	100%
f_2	76.88	100%	59.84	100%
g	36.24	100%	37.04	100%

	Two hidden layers: 8 and 4 neurons	
	Mean epoch number	Success rate
f_1	203.68	76%
f_2	135.54	96%
g	72.56	100%

Indeed, the learning of chaotic functions needs in average 284.57 epochs, whereas non chaotic functions require 232.87 epochs. In the future, we also plan to consider larger values for N .

7.2.2 Can a Neural Network Predict a Future Protein Conformation?

Experimental Protocol

In this second set of experiments, multilayer perceptrons are used to learn the conformation of very simple proteins (peptides, indeed). In fact, we consider proteins composed of five residues, of which only 4 can change since the first one is always 0, and folding dynamics of two or three folds. For example, if the current protein conformation is (0)1222, and folds 4 and -1 are successively applied, then the new conformation will be (0)0112. Obviously, these choices, that lead respectively to 20736 and 186624 potential conformations, do not correspond to realistic folding processes. However, they allow to evaluate the ability of neural networks to learn very simple conformations.

The networks consist of MLP with 3 or 4 inputs, the current conformation without the first residue, and a sequence of 2 or 3 successive folds. It produces a single output: the resulting conformation. Additionally, we slightly change the classical MLP structure in order to improve the capacity of such neural networks to model nonlinear relationships and to be trained faster. Therefore, we retain the HPU (Higher-order Processing Unit) structure [34]. This latter artificially increases the number of inputs by adding polynomial combinations of the initial inputs up to a given degree, called the order of the network. To prevent overfitting and to assess the generalization performance we use holdout validation, which means that the data set is split into learning, validation, and test subsets. These subsets are obtained through a random sampling strategy.

To estimate the prediction accuracy we use the coefficient of variation of the root mean square error (CVRMSE), usually presented as a percentage, the average relative variance (AVR), and the coefficient of efficiency denoted by (E).

These measures give a good estimation of the capacity of a neural network to explain the total variance of the data. The CVRMSE of the prediction is defined as:

$$\text{CVRMSE} = \frac{100}{\bar{e}_k} \cdot \sqrt{\frac{\sum_{k=1}^N (e_k - p_k)^2}{N}},$$

where e_k is the expected output for the k -th input-output pair, p_k is the predicted output, N is the number of pairs in the test set, and \bar{e}_k is the mean value of the expected output. The average relative variance and coefficient of efficiency are respectively expressed by:

$$\text{ARV} = \frac{\sum_{k=1}^N (e_k - p_k)^2}{\sum_{k=1}^N (e_k - \bar{e}_k)^2} \text{ and } E = 1 - \text{ARV}.$$

These values reflect the accuracy of the prediction as follows: the nearer CVRMSE and AVR to 0, and consequently E close to 1.0, the better the prediction.

Experimental Results

The considered neural networks differ in the size of a single hidden layer and are trained until a maximum number of epochs is reached. As we set the order of the HPU structure to 3, instead of 3 and 4 initial inputs we have 19 and 34 inputs. We train neural networks of 15 and 25 hidden neurons, using as maximum number of epochs a value in $\{500, 1000, 2500\}$, whatever the number of initial inputs (see Table 2). The learning, validation, and test subsets are built in such a way that they respectively represent 65%, 10%, and 25% of the whole data set. In the case of 3 initial inputs data sets of 5184, 10368, and 15552 samples are used, they represent 25%, 50%, and 75% of the 20736 potential conformations. For the 4 initial outputs case we restrict our experiments to a data set of 46656 samples, that corresponds to 25% of the 186624 potential conformations.

In Table 2 we give, for the different learning setups and data set sizes, the mean values of CVMSE, AVR, and E for the output. To compute these values, 10 trainings with random subsets construction and network parameters initialization were performed. It can be seen that in all cases the better performances are obtained for the larger networks (25 hidden neurons) that are the most trained (2500 epochs). Furthermore, a larger data set allows only a slight increase of the prediction quality. We observe for a three time increase of the data set: from 5184 to 15552 samples, a very small improvement of 1.66% for the coefficient E .

At a first glance, the prediction accuracy seems not too bad for the 3 initial inputs topology, with coefficients of efficiency above 0.9. However, remember that we try to predict very simple conformations far away from the realistic ones. Furthermore, a look at the results obtained for the second topology, the one with 4 initial outputs, shows that predicting a conformation that undergoes only one more folding transition is intractable with the considered learning setups: the efficiency coefficient is always below 0.5. Clearly, the different neural networks have failed at predicting the protein folding dynamics. A larger neural

network with a longer training process may be able to improve these results. But finding a learning setup well suited for the prediction of relevant proteins structures that are far more complex seems very hypothetical.

Finally, let us notice that the HPU structure has a major impact on the learning quality. Indeed, let us consider the coefficient of efficiency obtained for the data set of 5184 samples and a network composed of 25 hidden neurons trained during 2500 epochs. As shown in Table 2 the coefficient E is about 0.9285 if the neural network has an HPU structure of order 3, whereas experiments made without increasing the number of initial inputs give 0.6930 as mean value for E . Similar experiments in case of the second topology result in $E = 0.2154$ for the classical structure that has no HPU structure. That represents a respective decrease of more than 25 and 50%, from which we can say that MLP networks with a classical structure would have given worse predictions.

At this point we can only claim that it is not completely evident that compu-

Table 2: Results of the validation of networks with an HPU structure of order 3 for several numbers of hidden neurons

Topology	3 initial / 19 HPU inputs and 1 output			
Hidden neurons	Epochs	% CVRMSE	ARV	E
15 neurons	Data set of 5184 samples			
	500	24.97	0.3824	0.6176
	1000	20.67	0.2628	0.7372
	2500	16.69	0.1731	0.8262
	500	23.33	0.3373	0.6627
	1000	15.94	0.1565	0.8435
25 neurons	2500	10.75	0.0715	0.9285
	Data set of 10368 samples			
	500	26.27	0.4223	0.5777
15 neurons	1000	22.08	0.3000	0.7000
	2500	18.81	0.2225	0.7775
	500	24.54	0.3685	0.6315
25 neurons	1000	16.11	0.1591	0.8409
	2500	9.43	0.0560	0.9440
	Data set of 15552 samples			
15 neurons	500	24.74	0.3751	0.6249
	1000	19.92	0.2444	0.7556
	2500	16.35	0.1659	0.8341
	500	22.90	0.3247	0.6753
25 neurons	1000	15.42	0.1467	0.8533
	2500	8.89	0.0501	0.9499
	4 initial / 34 HPU inputs and 1 output			
15 neurons	Data set of 46656 samples			
	500	35.27	0.7606	0.2394
	1000	33.50	0.6864	0.3136
	2500	31.94	0.6259	0.3741
	500	35.05	0.7535	0.2465
	1000	32.25	0.6385	0.3615
25 neurons	2500	28.61	0.5044	0.4956

tational intelligence tools like neural networks are able to predict, with a good accuracy, protein folding. To reinforce this belief, tools optimized to chaotic behaviors must be found – if such tools exist. Similarly, there should be a link between the training difficulty and the “quality” of the disorder induced by a chaotic iteration function (their constants of sensitivity, expansivity, etc.), and this second relation must be found.

8 Conclusion

In this paper the topological dynamics of protein folding has been evaluated. More precisely, we have studied whether this folding process is predictable in the 2D model or not. It is achieved to determine if it is reasonable to think that computational intelligence tools like neural networks are able to predict the 3D shape of an amino acids sequence. It is mathematically proven, by using two different ways, that protein folding in 2D hydrophobic-hydrophilic (HP) square lattice model is chaotic according to Devaney.

Consequences for both structure prediction and biology are then outlined. In particular, the first comparison of the learning by neural networks of a chaotic behavior on the one hand, and of a more natural dynamics on the other hand, are outlined. The results tend to show that such chaotic behaviors are more difficult to learn than non-chaotic ones. It is not our pretension to claim that it is impossible to predict chaotic behaviors such as protein folding with computational intelligence. Our opinion is just that this important point must now be regarded with attention.

In future work the dynamical behavior of the protein folding process will be more deeply studied, by using topological tools as topological mixing, Knudsen and Li-Yorke notions of chaos, topological entropy, etc. The quality and intensity of this chaotic behavior will then be evaluated. Consequences both on folding prediction and on biology will then be regarded in detail. This study may also allow us to determine, at least to a certain extent, what kind of errors on the initial condition lead to acceptable results, depending on the intended number of iterations (i.e., the number of folds). Such a dependence may permit to define strategies depending on the type and the size of the proteins, their proportion of hydrophobic residues, and so on.

Other molecular or genetic dynamics will be investigate by using mathematical topology, and other chaotic behaviors will be looked for (as neurons in the brain). More specifically, various tools taken from the field of computational intelligence will be studied to determine if some of these tools are capable to predict behaviors that are chaotic with a good accuracy. It is highly possible that prediction depends both on the tool and on the chaos quality. Moreover, the study presented in this paper will be extended to high resolution 3D models. Impacts of the chaotic behavior of the protein folding process in biology will be regarded. Finally, the links between this established chaotic behavior and stochastic models in gene expression, mutation, or in Evolution, will be investigated.

References

- [1] M. Hoque, M. Chetty, A. Sattar, in *Biomedical Data and Applications, Studies in Computational Intelligence*, vol. 224, ed. by A. Sidhu, T. Dillon (Springer Berlin Heidelberg, 2009), pp. 317–342
- [2] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, M. Yannakakis, in *Proceedings of the thirtieth annual ACM symposium on Theory of computing* (ACM, New York, NY, USA, 1998), STOC '98, pp. 597–603
- [3] T. Higgs, B. Stantic, T. Hoque, A. Sattar, in *IEEE Congress on Evolutionary Computation* [35], pp. 1–8
- [4] A. Shmygelska, H.H. Hoos. An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem (2005 Feb)
- [5] L.G. Pérez-Hernández, K. Rodríguez-Vázquez, R. Garduño-Juárez, in *IEEE Congress on Evolutionary Computation* [35], pp. 1–8
- [6] M.K. Islam, M. Chetty, in *Proceedings of the 22nd Australasian Joint Conference on Advances in Artificial Intelligence* (Springer-Verlag, Berlin, Heidelberg, 2009), AI '09, pp. 412–421
- [7] I. Dubchak, I. Muchnik, S.R. Holbrook, S.H. Kim, *Proc Natl Acad Sci U S A* **92**(19), 8700 (1995)
- [8] C.B. Anfinsen, *Science* **181**(4096), 223 (1973). DOI 10.1126/science.181.4096.223. URL <http://www.sciencemag.org/content/181/4096/223.short>
- [9] R. Bonneau, D. Baker, *Annual Review of Biophysics and Biomolecular Structure* **30**(1), 173 (2001). DOI 10.1146/annurev.biophys.30.1.173
- [10] D. Chivian, D.E. Kim, L. Malmström, J. Schonbrun, C.A. Rohl, D. Baker, *Proteins* **61**(S7), 157 (2005). URL <http://dx.doi.org/10.1002/prot.20733>
- [11] Y. Zhang, A.K. Arakaki, J. Skolnick, *Proteins* **61**(S7), 91 (2005). URL <http://dx.doi.org/10.1002/prot.20724>
- [12] J. Bahi, C. Guyeux, N. Cote, in *IJCNN 2011, Int. Joint Conf. on Neural Networks* (San Jose, California, United States, 2011), pp. ***-***. To appear
- [13] G. Böhm, *Chaos, Solitons & Fractals* **1**(4), 375 (1991). DOI 10.1016/0960-0779(91)90028-8. URL <http://www.sciencedirect.com/science/article/B6TJ4-46CBXVT-1X/2/370489c218e4c2732cd9b620ef50c696>
- [14] H.b. Zhou, L. Wang, *The Journal of Physical Chemistry* **100**(20), 8101 (1996). DOI 10.1021/jp953409x

- [15] M. Braxenthaler, R.R. Unger, D. Auerbach, J. Moult, *Proteins-structure Function and Bioinformatics* **29**, 417 (1997). DOI 10.1002/(SICI)1097-0134(199712)29:4<417::AID-PROT2>3.3.CO;2-O
- [16] B. Berger, T. Leighton, in *Proceedings of the second annual international conference on Computational molecular biology* (ACM, New York, NY, USA, 1998), RECOMB '98, pp. 30–39
- [17] K. Dill, *Biochemistry* **24**(6), 1501 (1985). URL <http://ukpmc.ac.uk/abstract/MED/3986190>
- [18] M.K. Islam, M. Chetty, in *IEEE Congress on Evolutionary Computation* [35], pp. 1–8
- [19] R. Unger, J. Moult, in *Proceedings of the 5th International Conference on Genetic Algorithms* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993), pp. 581–588
- [20] D. Horvath, C. Chira, in *IEEE Congress on Evolutionary Computation* [35], pp. 1–8
- [21] H.J. Dyson, P.E. Wright, *Nature Reviews Molecular Cell Biology* **6**(3), 197 (2005). DOI DOI:10.1038/nrm1589
- [22] V.N. Uversky, C.J. Oldfield, A.K. Dunker, *Annual Review of Biophysics* **37**(1), 215 (2008). DOI 10.1146/annurev.biophys.37.032807.125924. PMID: 18573080
- [23] J.M. Bahi, C. Guyeux, *Journal of Algorithms & Computational Technology* **4**(2), 167 (2010)
- [24] A. Shmygelska, H. Hoos, *BMC Bioinformatics* **6**(1), 30 (2005). DOI 10.1186/1471-2105-6-30
- [25] R. Backofen, S. Will, P. Clote. Algorithmic approach to quantifying the hydrophobic force contribution in protein folding (1999)
- [26] R.L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd edn. (Addison-Wesley, Redwood City, CA, 1989)
- [27] J. Banks, J. Brooks, G. Cairns, P. Stacey, *Amer. Math. Monthly* **99**, 332 (1992)
- [28] J.M. Bahi, C. Guyeux, Q. Wang, in *ICCASM 2010, Int. Conf. on Computer Application and System Modeling* (Taiyuan, China, 2010), pp. V13–643–V13–647. DOI 10.1109/ICCASM.2010.5622199. URL <http://dx.doi.org/10.1109/ICCASM.2010.5622199>
- [29] J.M. Bahi, C. Guyeux, in *WCCI'10, IEEE World Congress on Computational Intelligence* (Barcelona, Spain, 2010), pp. 1–7. Best paper award

- [30] J. Bahi, J.f. Couchot, C. Guyeux, A. Richard, in *FCT'11, 18th Int. Symp. on Fundamentals of Computation Theory, LNCS*, vol. 6914 (Oslo, Norway, 2011), *LNCS*, vol. 6914, pp. 126–137
- [31] A. Andreeva, D. Howorth, S.E. Brenner, T.J.P. Hubbard, C. Chothia, A.G. Murzin, *Nucleic Acids Research* **32**(suppl 1), D226 (2004). DOI 10.1093/nar/gkh039
- [32] J. Bahi, C. Guyeux, M. Salomon, in *ICCANS 2011, IEEE Int. Conf. on Computer Applications and Network Security* (Maldives, Maldives, 2011)
- [33] K. Hornik, M.B. Stinchcombe, H. White, *Neural Networks* **2**(5), 359 (1989)
- [34] J. Ghosh, Y. Shin, *Int. J. Neural Syst.* **3**(4), 323 (1992)
- [35] *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July 2010* (IEEE, 2010)

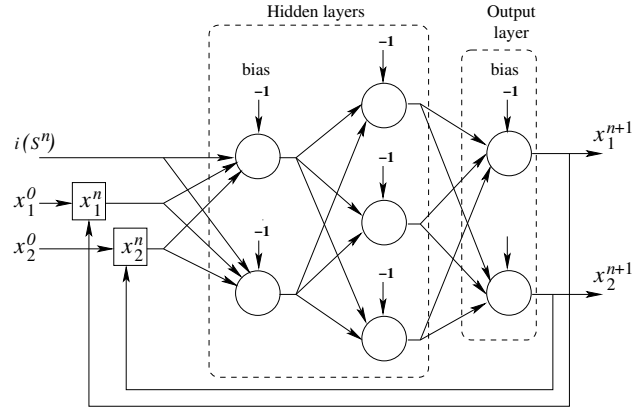


Figure 6: Recurrent neural network modeling F_f